Specification:

Replace the first paragraph on page 2 with the following rewritten paragraph:

This invention relates to Boolean functions that involve more than 100 Boolean variables, specifically to checking whether such a Boolean function is a tautology with a given constraint.

Replace the first full paragraph on page 3 with the following rewritten paragraph:

As digital circuits become more and more complex while integrated circuit technologies grow, equivalence checking methods and tautology checking methods often fail to handle the complex Boolean functions used to represent digital circuits. The failures are either due to unreasonably long run times or due to requiring unreasonably large amounts of computer memory. These methods (such as U.S. patent No. 5,243,538 and others using BDDs) give meaningful conclusions only at the end, and therefore they give no meaningful conclusions at all if they fail before reaching the end. A truth table of a Boolean function involving 100 Boolean variables can fill more than 1000 trillion storage devices if each of such storage devices can store 1000 trillion rows of the truth table, which is much larger than most modern hard disks or tapes. Just touching such amount of data at the speed of 30 billion rows per second (faster than the modern speed of accessing registers) takes more

than a trillion years per pass. It is well known that BDDs of many practical Boolean functions have sizes similar to these truth tables'.

Replace the last paragraph on page 9 with the following rewritten paragraph:

There are many ways to represent a Boolean function in a computer for a program to process. Because the representation of Boolean functions is only involved in the simplification and the constant substitution within steps **120** and **220**, this invention can use any computer-oriented representation that is concise and fast to build for complex Boolean functions that involve more than 100 Boolean variables. Plain truth tables are not good for this purpose because they take too much memory (due to the more than 1000 trillion rows) for more than 50 Boolean variables. Binary decision diagrams are not good for this purpose because they sometimes have sizes similar to plain truth tables, and are not fast to build in many cases. The preferred embodiment uses ~~directed acyclic graphs~~netlists to represent Boolean functions. A netlist may also be called a logic circuit or a directed acyclic graph. Each node of the ~~directed acyclic graph~~netlist is a circuit block (a logic gate, a Boolean expression, a truth table, etc.). As well known in the art, a ~~directed acyclic graph~~netlist is the natural representation of an interconnected collection of circuit components. Accordingly, steps **120** and **220** also have to use computer-oriented simplification algorithms. The simplification algorithms in the preferred embodiment include constant propagation.

Replace the last paragraph on page 12 with the following rewritten paragraph:

With these flexible division techniques, very large Boolean functions can be handled in tautology checking even if such a Boolean function involves much more than 100 Boolean variables. This is required for automatically verifying large data-processing systems using computers, especially when verifying them against the expected behaviors without any assumptions of their internal structures.
